International Journal of Electronics, Communication & Instrumentation Engineering Research and Development (IJECIERD)
ISSN(P): 2249-684X; ISSN(E): 2249-7951
Vol. 4, Issue 1, Feb 2014, 37- 46

TRANS STELLAR
• Journal Publications • Research Consultancy

DESIGN AND IMPLEMENTATION OF HIGH SPEED SIGNED MULTIP LIER USING 3_2 COMPRESSOR

D. SRINU[1], S. RAMBABU[2] & G. LEENENDRA CHOWDARY[3]

[1]Research Scholar, Department of ECE, SITE, Tadepaliigudem, Andhra Pradesh, India

[2,3]Assistant Professor, Department of ECE, SITE, Tadepaliigudem, Andhra Pradesh, India

ABSTRACT

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area for compact VLSI implementation. Multiplier is based on the ancient algorithms (sutras) for multiplication [1]. This work is based on one of the sutras called *Urdhava Tiryakbhyam*. These sutras are meant for faster mental calculation. Though faster when implemented in hardware, it consumes less area. This paper presents a technique to modify the architecture of the *Urdhava Tiryakbhyam* by using compressor in order to reduce area and delay to improve overall performance. The coding is done for 16 bit(Q15), 32 bit(Q31) and 64 bit(Q63) fractional fixed point multiplications using Verilog HDL and Synthesized using Xilinx ISE version 9.2i. The performance is compared in terms of area, delay with earlier existing architecture of Vedic multiplier. The proposed design shows very good improvements in terms of area and time delay.

KEYWORDS: Compressor Fractional Fixed Point Format Q-Format, Urdhava Tiryakbhyam, Vedic Multiplier

INTRODUCTION

Vedic Mathematics is the ancient system of mathematics which was rediscovered early last century by Sri Bharati Krisna Tirthaji [1]. The Sanskrit word "Veda" means "knowledge".He organized and classified the whole of Vedic Mathematics into 16 formulae or also called as *sutras.* These formulae form the backbone of Vedic mathematics. Great amount of research has been done all these years to implement algorithms of Vedic mathematics on digital processors.

Currently implemented in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform (FFT), filtering and in microprocessors in its arithmetic and logic unit [3]. For multiplication algorithms performed in DSP applications latency and throughput are the two major concerns from delay perspective.

The algorithm to architecture mapping using floating point number representation consumes more hardware which tends to be expensive to overcome

This drawback we chose for fixed point number representation is a good option to implement at silicon level [2]. Hence our focus in this work is to develop optimized hardware modules for multiplication operation considering fixed point representation, 16 bit Q15 format, 32 bit Q31 format and 64 bit Q63 format provide required precision for most of the digital signal processing applications and it is best suited for implementation on processors. In this paper we propose the implementation of fixed point Q-format [6] high speed multiplier using compressed Urdhava Tiryakbhyam method of Vedic mathematics. results clearly shows that compressed Urdhava Tiryakbhyam method is best suited for implementing multipliers.

FIXED POINT ARITHMETIC

An N-bit fixed-point number can be interpreted as either: an integer (i.e., 20645), a fractional number (i.e., 0.75) N-bit fixed point, 2's complement integer representation

$$x=-b_{n-1}2^{n-1}+b_{n-2}2^{n-2}+\text{----------}+b_0 2^0$$

Integer fixed point is difficult to use in processors due to possible overflow In a 16-bit processor dynamic range in between -32,768 to 32,767.

Example

200 × 350 = 70000, this is an overflow! To overcome these draw back Fractional Fixed-Point Representation will be used which is suitable for DSP algorithms. Fractional number range is between 1 and -1 Multiplying a fraction by a fraction always results in a fraction and will not produce an overflow (e.g., 0.99 x 0.9999 less than 1) Successive additions may cause overflow Represent numbers between -1.0 and $1-2^{-n-1}$ when N is number of bits

Q- Format Representation

In general any Q-format representation is denoted by Q*m.n notation* m bits for integer portion, n bits for fractional portion Total number of bits N = m + n + 1, for signed numbers

Example: 16-bit number (N=16) and Q2.13 format 2 bits for integer portion, 13 bits for fractional portion, 1 signed bit (MSB)

Special cases: 16-bit integer number (N=16) => Q15.0 format 16-bit fractional number (N = 16) => Q0.15 format; also known as Q.15 or Q15

Q-Format Multiplication

When two Q15 numbers are multiplied their product is 32 bits long as illustrated in Figure 1. The product has a redundant or extended sign bit. Since the product stored in memory should also be a Q15 number we left shift the product by one bit and the most significant 16 bits (including sign bit) is stored in the memory

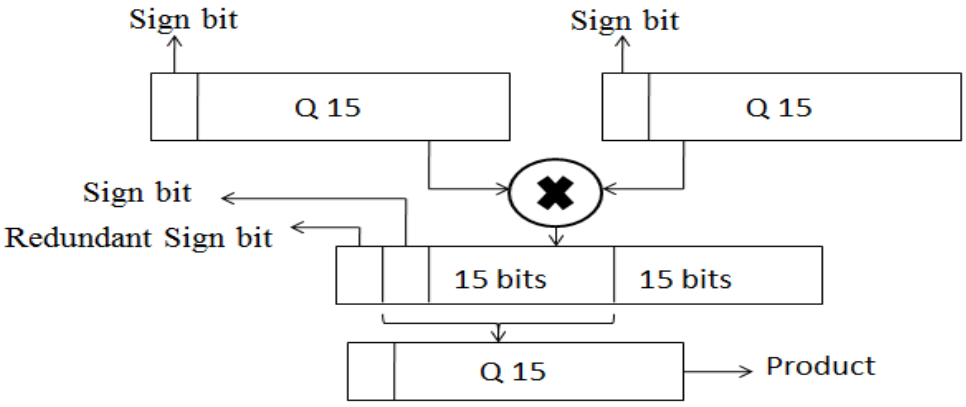

Figure 1: Multiplications of Two Q15 Format Numbers Yielding the Product in Q15 Formats itself

Product of two Q15 numbers is Q30.So we must remember that the 32-bit product has *two bits* in front of the binary point. Since NxN multiplication yields 2N-1 result Addition MSB sign extension bit typically only the most significant 15 bits (plus the sign bit) are stored back into memory, so the *write operation requires a left shift by one*.

URDHAVA TIRYAKBHYAM METHOD

Urdhava Tiryakbhyam [2] is a Sanskrit word which means vertically and crosswire in English. The method is a general multiplication formula applicable to all cases of multiplication. It is based on a novel concept through which all partial products are generated concurrently.

Figure 2 demonstrates a 4 x 4 binary multiplication using this method. The method can be generalized for any N x N bit multiplication. This type of multiplier is independent of the clock frequency of the processor because the partial products and their sums are calculated in parallel. The net advantage is that it reduces the need of microprocessors to operate at increasingly higher clock frequencies. As the operating frequency of a processor increases the number of switching instances also increases. This results in more power consumption and also dissipation in the form of heat which results in higher device operating temperatures. Another advantage of Urdhva Tiryakbhyam multiplier is its scalability. The processing power can easily be increased by increasing the input and output data bus widths since it has a regular structure. Due to its regular structure, it can be easily layout in a silicon chip and also consumes optimum area. As the number of input bits increase, gate delay and area increase very slowly as compared to other multipliers. Therefore Urdhava Tiryakbhyam multiplier is time, space and power efficient.

Figure 2, the least significant bit (LSB) of the multiplier is multiplied with least significant bit of the multiplicand (vertical multiplication). This result forms the LSB of the product. In step 2 next higher bit of the multiplier is multiplied with the LSB of the multiplicand and the LSB of the multiplier is multiplied with the next higher bit of the multiplicand (crosswire multiplication). These two partial products are added and the LSB of the sum is the next higher bit of the final product and the remaining bits are carried to the next step the partial products and their sums for every step can be calculated in parallel.

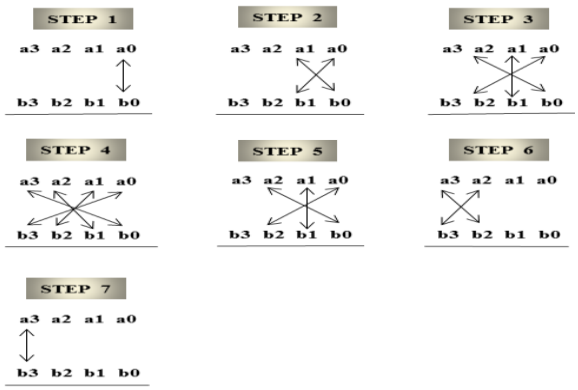

Figure 2: Multiplications of Two 4 Bit Numbers Using Urdhava Tiryakbhyam Method. [7]

Thus every step in Figure 2 has a corresponding expression as

Follows:

r0=a0b0. (1)

c1r1=a1b0+a0b1. (2)

c2r2=c1+a2b0+a1b1 + a0b2. (3)

c3r3=c2+a3b0+a2b1 + a1b2 + a0b3. (4)

c4r4=c3+a3b1+a2b2 + a1b3. (5)

c5r5=c4+a3b2+a2b3. (6)

c6r6=c5+a3b3 (7)

With c6r6r5r4r3r2r1r0 being the final product [5].

ARCHITECTURE

Our design of Q-format signed multiplier includes Urdhava Tiryakbhyam integer multiplier [4] with certain modifications as follows

A. 3_2 Compressor

High speed multipliers use 3-2, 4-2 and 5-2 compressors to lower the latency of partial product reduction part [8]. These compressors are used to minimize delay and area which leads to increase the performance of the overall system. Compressors are generally designed by XOR-XNOR gates and multiplexers. A compressor is a device which is used to reduce the operands while adding terms of partial products in multipliers. An X-Y compressor takes X equally weighted input bits and produces Y-bit binary number. The most widely and the simplest used compressor is the 3-2 compressor which is also known as a full adder. A 3-2 compressor has three inputs X1, X2, X3 and generates two

Outputs they are sum and the carry bits. The block diagram of 3-2 compressor is shown in figure.

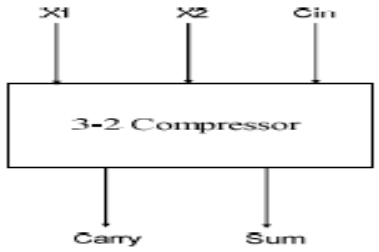

Figure 3: (a) A 3_2 Compressor

X1	X2	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 3: (b) 3_2 Compressor Truth Table

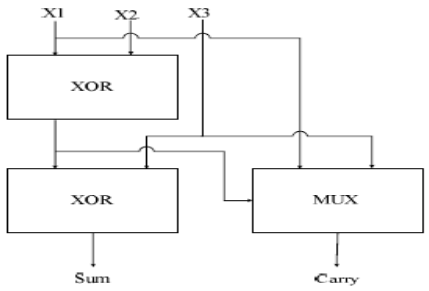

Figure 3: (c) Conventional 3_2 Compressor

The conventional architectures of 3-2 compressor shown in figure 3(c), it has two XOR gates in the critical path.

The sum output is generated by the second XOR and carry output is generated by the multiplexer (MUX). The Equations governing the conventional 3-2 compressor outputs are shown below:

X1+X2+X3=sum+2●carry---------- (1)

Sum=X1 ⊕X2⊕X3---------- (2)

Carry= (X1⊕X2) ●$\overline{X3}$+ (X1⊕X2) ●X1---------- (3)

This compressor based multiplier is faster since all the partial products are computed concurrently. Considering a 16 bit Q15 multiplier, the product is also a Q15 number which is 16 bits long. Firstly, if the MSB of input is 1 then it is a negative number. Therefore 2's complement of the number is taken before proceeding with multiplication. Since the MSB denotes sign it is excluded and a '0' is placed in this position while multiplying. A Q15 format multiplier consists of four 8 x 8 Urdhava multipliers and the resulting product is 32 bits

Long as shown in Figure 4 But the product of a Q15 number is also a Q15 number which should be 16 bits long.

Therefore the 32 bit product is left shifted by 1 bit to remove the redundant sign bit and only the most significant 16 bits of this product are considered which constitute the final product. An xor operation is performed on the input sign bits to determine the sign of the result. if the output is '1'it enables the conversion of the 16 bit final result to its 2's compliment format indicating a negative product.

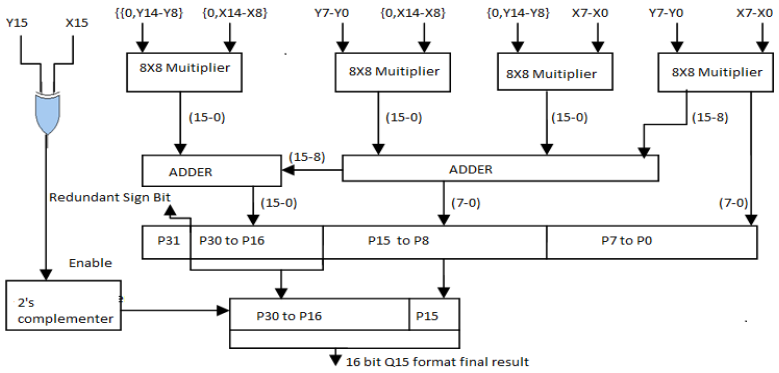

Figure 4: Architecture of a Q15 Format Multiplier. Multiplication of Two Q15 Numbers X and Y Results in a Q15 Product Denoted by P in the Figure

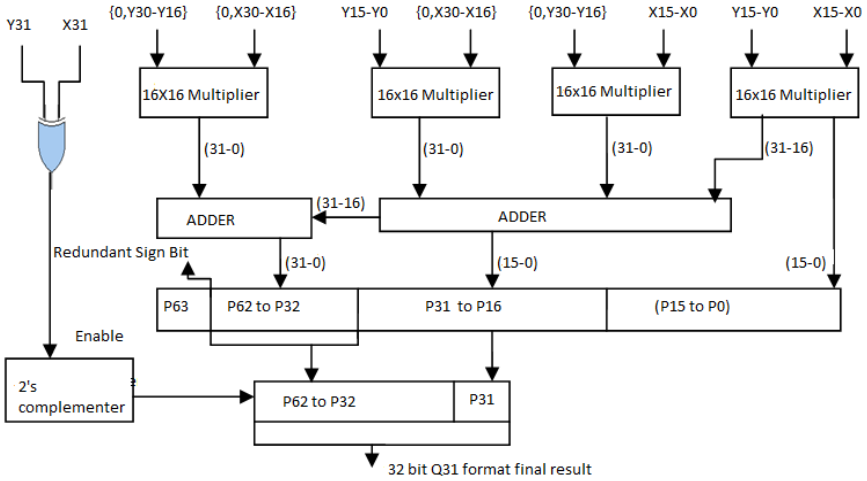

Figure 5: Architecture of a Q31 Format Multiplier. Multiplication of Two Q31 Numbers X and Y Results in a Q31 Product Denoted by P in the Figure

In Figure 5 but the product of a Q31 number is also a Q31number which should be 32 bits long. Therefore the 64 bit product is left shifted by 1 bit to remove the redundant sign bit and only the most significant 32 bits of this product are considered which constitute the final product. A xor operation is performed on the input sign bits to determine the sign of the result. If the output is '1'it enables the conversion of the 32 bit final result to its 2's compliment format indicating a negative product.

Similarly as shown in Figure 6 but the product of a Q63 number is also a Q63 number which should be 64 bits long. Therefore the 128 bit product is left shifted by 1 bit to remove the redundant sign bit and only the most significant 64 bits of this product are considered which constitute the final product. A xor operation is performed on the input sign bits to determine the sign of the result. If the output is '1'it enables the conversion of the 64 bit final result to its 2's compliment format indicating a negative product.

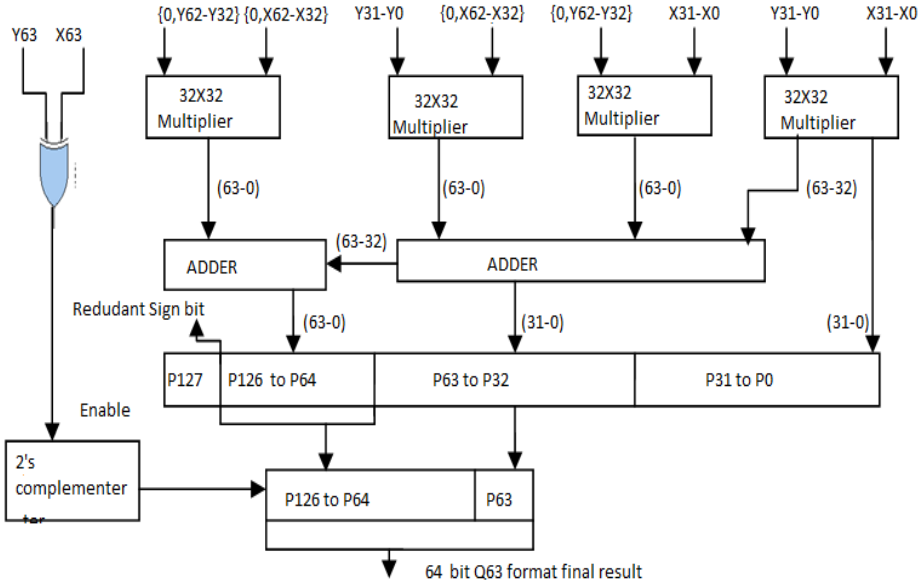

Figure 6: Architecture of a Q63 Format Multiplier. Multiplication of Two Q63 Numbers X and Y Results in a Q63 Product Denoted by P in the Figure

IMPLEMENTATION AND RESULTS

The proposed compressed Urdhava tirykbhyam Q_format multiplier is designed using verilog HDL and structural form of coding. The basic block of both Q15 and Q31 multiplier is a 4 x 4 Urdhava Tiryakbhyam integer multiplier which in turn is made up of two 2 x 2 multiplier blocks.. The Code is completely synthesized using Xilinx XST and Implemented on device family Virtex-5, device XC5VL50, Package FF324 with speed grade -2.

Simulation Results

The design was simulated using Isim on Xilinx ISE 9.2i version.

For Q15 format multiplication as shown in Figure 6,

Input1 =-0.75 = 1010 0000 0000 0000

Input2 =- 0.25 = 1100 0000 0000 0000

Output = 0.1875 = 0001 1000 0000 0000

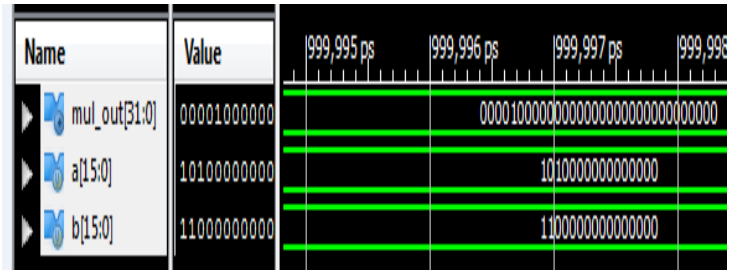

Figure 7: Q15 Multiplication

For Q31 format multiplication as shown in Figure 8.

Input1=0.333333 = 00101010101010101010011111011111

Input2=-0.666666 = 10101010101010101011000001000010

Output= -0.22222177777439355850219726556 25= 1111

1000 1110 0011 0001 1011 1000 0101 But the actual value of the product is -0.222221777778. Therefore precision loss is involved in this Multiplication and is found to be 3.60644E-12 which is less Than the resolution of Q31 representation i.e. 2-31. Thus it Provides 32 bit accurate product which is acceptable for most Of the DSP applications.

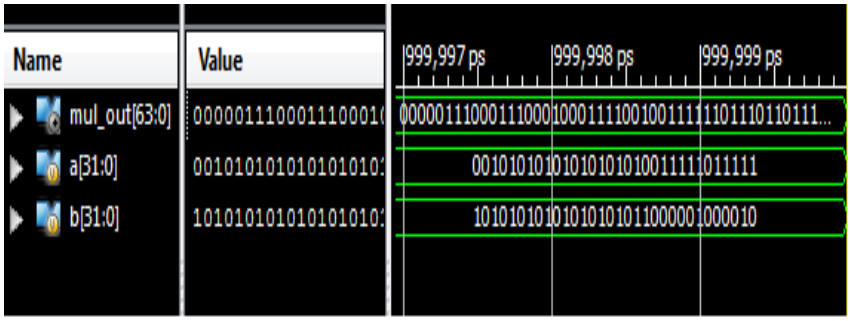

Figure 8: Q31 Multiplication

For Q63 format multiplication as shown in Figure 8.

Input1= 0000111111110000111100001111000011110000111100001111000011110000111100 0011110000

Input 2= 1010101010101010101010101010101010101010101010101010101010101010 1010101010

Output=

1010000010100000101000001010000010100000101000001010000010011111010111110101111101011111010111110101111101
0111110101111101011111010111110101111101100000

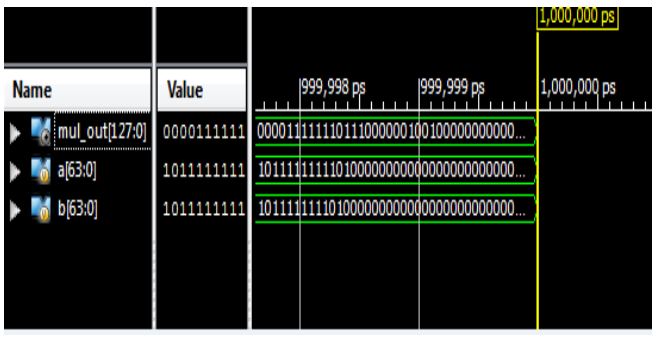

Figure 9: Q63 Multiplication

RTL SCHEMATICS

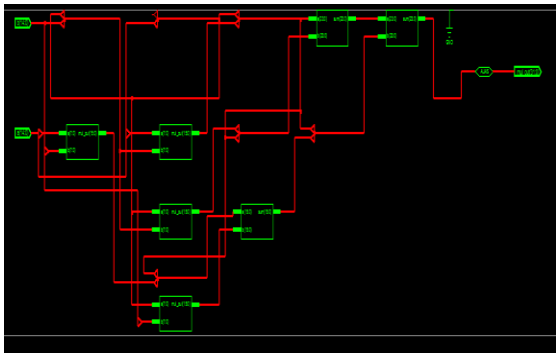

Figure 10: RTL Schematic of 16 BIT

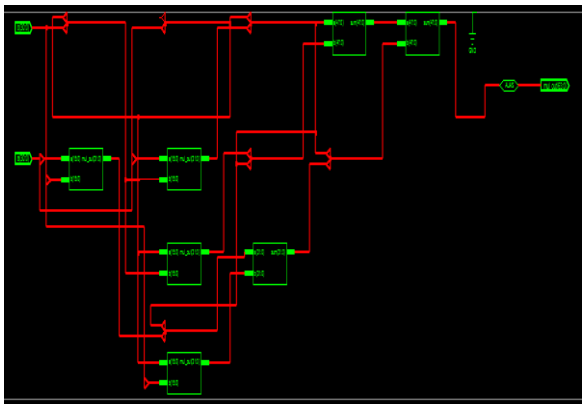

Figure 11: RTL Schematic of 32 BIT

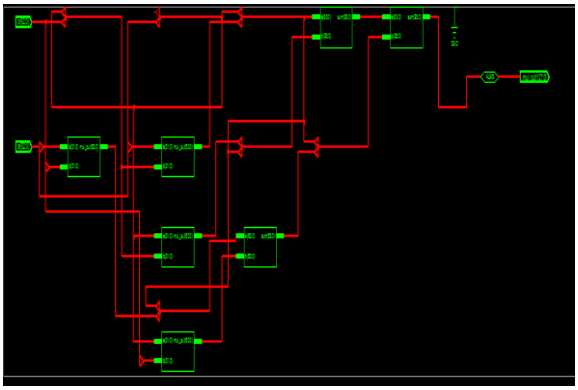

Figure 12: RTL Schematic of 64 BIT

Table 1: Comparison of Area Occupied and Speed of Various Multiplier Architectures for 16 Bit

Algorithm used	LUTs Used	Total LUTs present	%of area occupied	Frequency (MHz)	Time (ns)
Urdhwa-Tiryakbhyam	425	28800	1.47	70.24	14.236
Compressor based Urdhwa Tiryakbhyam	421	28800	1.46	77.88	12.840

Table 2: Comparison of Area Occupied and Speed of Various Multiplier Architectures for 32 Bit

Algorithm used	LUTs Used	Total LUTs present	%of area occupied	Frequency (MHz)	Time (ns)
Urdhwa-Tiryakbhyam	1804	28800	6.26	55.93	17.879
Compressor based Urdhwa Tiryakbhyam	1787	28800	6.20	60.48	16.483

Table 3: Comparison of Area Occupied and Speed of Various Multiplier Architectures for 64 Bi

Algorithm used	LUTs Used	Total LUTs present	%of area occupied	Frequency (MHz)	Time (ns)
Urdhwa-Tiryakbhyam	7432	28800	25.80	45.34	22.054
Compressor based Urdhwa Tiryakbhyam	7363	28800	25.56	47.93	20.863

CONCLUSIONS

This paper proposed fast multiplier architecture for signed Q-format multiplications using compressor based Urdhava Tiryakbhyam method of Vedic mathematics. Since Q-format representation is widely used in Digital Signal Processors. the proposed compressed Urdhava Tiryakbhyam method can substantially speed up the multiplication operation which is the basic hardware block. They occupy less area and are faster than the Urdhava Tiryakbhyam method. Therefore the compressed Urdhava Tiryakbhyam Q-format multiplier is best suited for digital signal processing applications requiring faster multiplications.

REFERENCES

1. Jagadguru Swami Sri Bharati Krisna Tirthaji Maharaja, "*Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Veda*," Motilal Banarasidas Publishers, Delhi, 2009, pp. 5-45.
2. H. Thapliyal and M. B. Shrinivas and H. Arbania, "*Design and Analysis of a VLSI Based High Performance Low Power Parallel Square Architecture,*" Int. Conf. Algo.Math.Comp. Sc., Las Vegas, June 2005, pp. 72-76.0
3. Sandesh S. Saokar, R. M. Banakarand Saroja Siddamal**,"** High Speed Signed Multiplier for Digital Signal Processing Applications" 978-1-4673-1318-6/12©2012 IEEE
4. M. Pradhan and R. Panda, "*Design and Implementation of Vedic Multiplier,*" A.M.S.E Journal, Computer Science and Statistics, France vol. 15, July 2010, pp. 1-19.

5. Harpreet Singh Dhillon, Abhijit Mitra, "*A Reduced-Bit Multiplication Algorithm for Digital Arithmetic's,*" International Journal of Computational and Mathematical Sciences, spring 2008, pp.64-69.

6. Sen-Maw Kuok and Woon-Seng Gan, "*Digital Signal Processor, architectures, implementations and applications,*" Pearson PrenticeHall, 2005, pp. 253-323.

7. S.S. Kerur, Prakash Narchi, Jayashree C. N. Harish M.Kittur and GirishV.A. "*Implementation of Vedic Multiplier for Digital Signal Processing*," International Journal of Computer Applications, 2011, vol. 16, pp. 1-5.

8. Jorge Tonfat, Ricardo Reis, "Low Power 3-2 and 4-2 Adder Compressors Implemented Using ASTRAN",

9. *IEEE Third Latin American Symposium on Circuits and Systems* (LASCAS), Feb. 29, March 2, 2012.